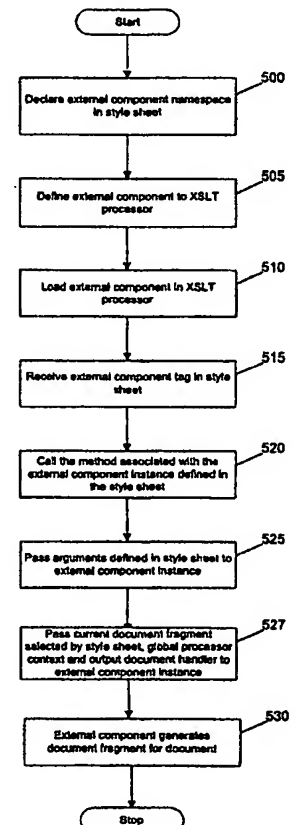


PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 17/21, 17/22, 17/30	A1	(11) International Publication Number: WO 00/54174 (43) International Publication Date: 14 September 2000 (14.09.00)
(21) International Application Number: PCT/US00/06379 (22) International Filing Date: 13 March 2000 (13.03.00) (30) Priority Data: 60/123,916 12 March 1999 (12.03.99) US Not furnished 10 March 2000 (10.03.00) US (71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, MS PAL01-521, Palo Alto, CA 94303 (US). (72) Inventor: YALCINALP, L., Umit; 1 Debby Lane, Belmont, CA 94002 (US). (74) Agents: GARRETT, Arthur, S.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US) et al.		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>
(54) Title: EXTENDING THE CAPABILITIES OF AN XSL STYLE SHEET TO INCLUDE COMPONENTS FOR CONTENT TRANSFORMATION (57) Abstract <p>Systems and methods consistent with the present invention use a Namespace paradigm to define an external component reference to a style sheet (500). When the style sheet processor processes the tags in the style sheet, it recognizes the external component declaration (510). The style sheet will contain a name of the external component instance and a definition of the method to execute associated with the external component instance which is executing (520). The XSLT processor then relinquishes control to the external component to execute the method defined in the style sheet (525). The results of the method's execution may be placed in the transform document generated by processing the style sheet (530).</p>		



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

EXTENDING THE CAPABILITIES OF AN XSL STYLE SHEET TO INCLUDE
COMPONENTS FOR CONTENT TRANSFORMATION

RELATED APPLICATIONS

Provisional U.S. Patent Application No. 60/123,916 entitled "Enhanced XML
5 Processing for Content Transformation" filed March 12, 1999, is relied upon and is
incorporated by reference in its entirety in this application.

BACKGROUND

Field of the Invention

The present invention relates to the use of style sheets in creating a document,
10 and more particularly, to the use of external components in an XSL style sheet.

Description of the Related Art

Systems' and applications' use of documents has become so prolific that style
sheets are now often used to help manage the documents' display. Style sheets
provide greater flexibility and control over the display of a document's content. XSL
15 style sheets also allow the content of documents to be transformed, making them as
document transformers where the resulting documents may or may not be used for
display. A user may request a document and the application associated with the
document will use the information contained in the style sheet to display a new
document incorporating the information contained within the style sheet and the
20 requested document.

Typically, an XSL style sheet includes the use of tags. Tags are codes that
identify an element in the document, such as heading or font, for the purpose of
formatting the document. For example, the tag "<BOLD>" indicates that the text
associated with this tag should appear in bold. Style sheets also, however, have
25 become more advanced, to include a type of programming language. An example of
this is an Extensible Style Language (XSL) style sheet. XSL is a declarative style
sheet language specified in Extensible Markup Language (XML) which can also be
used to transform XSL documents. The XSL is actually more analogous to a
programming language than to a mechanism designed purely to analyze tags and
30 display attributes.

With XML, developers may provide functionality by creating their own customized tags. For example, XML supports links that point to multiple documents, as opposed to HTML links, which can reference just one destination each.

XML documents may be served to different clients with varied interests and capabilities. For example, a PC running NETSCAPE may require a document formatted differently than a PDA would. XSL is the style language used by XML to allow different clients to receive different XML documents. XSL is a specification language for separating style from content when creating XML or HTML pages. XSL style sheets allow a single style to be applied to multiple documents.

However, there are limitations to the use of style sheets. An application utilizing a style sheet to display a document often requires that the information contained in the style sheet be application dependent. In other words, the application must be aware of all the tag definitions. In addition, style sheets typically are not very robust and while they use a programming language, do not provide for external calls to components or libraries which may be used to aid in the modification of document information to be displayed.

Similarly, the use of XSL style sheets has been limited. Information contained within the XSL style sheet must be defined within the application and the XSL style sheet itself, and therefore the style sheet becomes application dependent. This sometimes is not desirable when using a style sheet across multiple applications, in which case a separate style sheet for each application would need to be developed. Furthermore, the XSL style sheet is completely self-contained in that no modules or functions can be called outside the style sheet itself. Therefore, what is needed in the art is the ability to create a style sheet that is application independent and to provide a means for creating application dependent functions within that style sheet.

SUMMARY OF THE INVENTION

Methods, systems and articles of manufacture consistent with the present invention overcome the shortcomings of existing style sheets by providing the ability to define components in a style sheet in order to execute methods outside the application. Using a Namespace paradigm, these components, declared within the style sheet, will be recognized by a style sheet processor.

In accordance with one aspect of the present invention, as embodied and broadly described herein, in a data processing system, a method of creating a transform document using a style sheet comprises the steps of receiving a request for an input document, retrieving the style sheet, having tags, associated with the input document, wherein one of the tags represents an external component, processing the tags, including one tag representing an external component, in the style sheet to generate a transform document, and processing a method associated with the external component. An additional step of placing the results of processing the method associated with the external component in the transform document may be performed. The method associated with the external component may generate application specific results. In addition, the style sheet may contain commands written in an XSL format. The step of processing the method associated with the external component may include loading the external component in an XSLT processor and initiating the execution of the method associated with the external component. In addition, the step of validating the style sheet and the input document may be performed. And finally, the step of locating an existing instance of the external component, wherein the processing of the one tag representing an external component includes using the existing instance, may be performed.

In accordance with another aspect of the present invention, as embodied and broadly described herein, in a data processing system, a method for executing an external component in a style sheet, comprising the steps of defining an external component to a style sheet processor, providing a definition of the external component in the style sheet, and processing the external component by the style sheet processor. The steps of loading the external component into the style sheet processor, calling the method associated with the external component defined in the style sheet, and passing an argument contained in the style sheet to the external component for use by the method may also be performed. In addition, the step of loading the external component including the step of locating an existing instance of the external component may also be performed. The external component may be defined to an XSLT processor and the style sheet using a Namespace. And, an application associated with an input document using the style sheet may not recognize the

external component. Processing the external component may include executing a method associated with the external component which generates a result, and placing the result in a transform document.

5 In accordance with yet another aspect of the present invention, as embodied and broadly described herein, a system for processing external components in a style sheet comprises a style sheet containing a definition of an external component and a definition of a method associated with the external component, and a style sheet processor operative to receive a request for an input document, retrieve the style sheet, and call the method associated with the external component defined in the style sheet.

10 The style sheet may contain an argument to pass to the method associated with the external component. In addition, a Namespace may be used in defining the external component to the style sheet and the style sheet processor. The style sheet processor may be an XSLT processor and the style sheet contains tags written in an XSL format. And finally, the style sheet processor may be further operative to generate a transform

15 document and to insert a result of the call to the method associated with the external component.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention.

20 In the drawings,

Fig. 1 depicts a data processing system suitable for practicing methods and systems in a manner consistent with exemplary embodiments of the present invention;

25 Fig. 2 depicts a more detailed view of an XSLT processor and components related to the XSLT processor suitable for practicing methods and systems in a manner consistent with an exemplary embodiment of the present invention;

Fig. 3 is a flowchart depicting a high level process of generating a transform document using the results of an external call contained within a style sheet;

Fig. 4 is a flowchart depicting the process for generating a transform document containing the result of executing a method associated with an external component; and

Fig. 5 is a method for creating and processing external components in a style sheet.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The following detailed description of the invention refers to the accompanying drawings. Although the description includes exemplary implementations, other implementations are possible, and changes may be made to the implementations described without departing from the spirit and scope of the invention. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Introduction

Systems and methods consistent with the present invention use a Namespace paradigm to define an external component to a style sheet. Namespaces are covered in more detail by the "Namespaces in XML" recommendation specification by the World Wide Web Consortium. When the style sheet processor processes the tags in the style sheet, it recognizes the external component declaration and a locator to load the component. The style sheet will also contain a definition of the method to execute associated with the external component and a name which designates a particular instance of a component, and may contain arguments for the method which is executing. The style sheet processor, or the XSLT processor in one embodiment, then relinquishes control to the named external component instance to execute the method defined in the style sheet. The results of the method's execution may be placed in the new document generated by processing the style sheet. However, this is not required, and this execution may provide any results including those not related to the document.

the user request how to format the transformed document specific to the client specification.

5 The method then proceeds to generate a transform document using the style sheet and an external call embedded in an external component in the style sheet (step 305). The external call is to a named component instance and may perform a function defined by the developer creating the external call. In other words, it is not required that the external call perform some function related to the transformed document or even a function related to the document processing. It may be preferable for the developer of the style sheet to insert into the external function parameters and
10 information that may be application dependent. However, as stated above, the external call may also perform various other types of processing.

After processing the external call embedded in the external component, the XSLT processor continues to process the rest of the style sheet and the input document. Other methods of external components embedded within the style sheet
15 might be activated as described above. When the style sheet is completely processed by the XSLT processor, the process sends a transform document containing the results of the external call to the user (step 310). The new document will be transformed or formatted using commands contained in a style sheet associated with the document requested by the user. For example, these commands may define display parameters and other aspects associated with the new document display on the client type for the
20 user. In addition, the style sheet may contain the definition of the external call, as well as any arguments that may be passed through the external call.

Figure 4 is a flowchart depicting the process for generating a transform document containing the result of processing an external component in a style sheet.
25 The process begins by receiving a document request from a user (step 400). The user may request this by using a PC or other type of data processing system by which he has access to the XSLT processor. The XSLT processor receives the document request and client type from the user and validates the document and the style sheet associated with the document (step 405). As part of the validation process, in one
30 embodiment, the XSLT processor validates a document by using an XML parser and

embodiment of the invention, a method will be defined to the external component. The process proceeds by executing the method in the external component instance (Step 415). As part of executing this method, various parameters, i.e., arguments, may be associated with the component that the user has specified in the style sheet.

5 These parameters and the document fragment which is currently being processed by the style sheet will be passed to the external component that is being activated. The XSLT processor, when processing the style sheet, relinquishes control to the instance of the component so that it performs the method defined in the style sheet. After that method has been completed, processing returns to the XSLT processor to continue

10 processing the style sheet and hence the input document.

Once all the tags in the style sheet have been executed, as well as any other method(s) associated with external components contained in the style sheet, the XSLT processor then places in a transform document the results of the tag processing as well as the results of the executed method(s) (step 420). This transform document is now

15 transformed or formatted as desired by the user. It may contain the display characteristics or the transformed document content and tags as intended by the user, as well as the results of any external component processing as defined in the style sheet. Following the placement of the results of the tag processing, including external components, this new document is transmitted to the user using the associated client

20 type (step 425).

Figure 5 is a method for creating and processing external components in a style sheet. This process begins by starting to validate a style sheet and declaring the component Namespace in the style sheet (step 500). This may be performed by defining it as a specific Namespace in the style sheet. The method of declaring

25 Namespaces is well known to those skilled in the art. However, in this instance, the Namespace being declared is a Namespace defining external component activation in a style sheet.

Next, during the validation of the style sheet and processing of the XSLT tags, an external component identified by the specific Namespace declaration is identified

30 and is defined to the XSLT processor (step 505). The use of Namespaces is well known to those skilled in the art. The Namespace paradigm may be used by an

Next, the method associated with the component is called (step 522). Any values defined in the style sheet for the method received by the XSLT processor are then passed to the specific named method (step 525) as the arguments of the method in the component. The specific method will use the values passed from the XSLT processor as its arguments. This step can use the methods specified in the named external component.

A global context which the XSLT processor passes to the components to utilize and the current document fragment being processed by the style sheet and a transformed document fragment generator are also supplied to the component (Step 527). The methods in the external component instance take all the argument values specified in the method invocation. Three pieces of information need to be available for the external component instance: 1) the document fragment that is currently being processed as selected by the style sheet; 2) the global context that is provided by the XSLT processor; and 3) a means for creating document fragments in the transformed document that is provided to the component instance. The component instance may use or not use the value of the input document fragment or the context during its processing, but should depend on the availability of these items provided by the processor during runtime. This can be handled by different implementation techniques, either by requiring those three additional items to be included in the method signatures by allowing the XSLT processor to pass them to the component instance, or by requiring the component to use a specific API as well as its own API which is specialized with three different methods that pertain to obtaining these three items.

The invention also suggests a specific API which allows the component to obtain the document fragment, means of generating the resulting document fragment, and a global processing context which can be shared among components and their instances. Components which can be activated with this paradigm may utilize an API to be aware of the document processing and the context. For example, a getInputNode API may be used to get an input document fragment, a getDocumentFactory API may be use to create output document fragment and a getContext API may be used global context processing. At runtime, the components

When the XSLT processor completes the processing of the style sheet, the new document containing the results of the style sheet along with the results of all external component instances is placed in the new document and sent to the user. It will be recognized by those skilled in the art that while the foregoing description included a discussion of one external component being executed in the style sheet, multiple external components may be executed and defined in a style sheet as defined by the user.

An Example of an External Component in a Style Sheet

Below is an example of a style sheet written in XSL:

```

10 <?xml version="1.0"?>
    <!--XSL Style sheet, DTD omitted -->
    <xsl:stylesheet
        xmlns:xsl="http://www.w3.org/TR/WD-xsl"
        xmlns:xslcomponent="http://www.javasoft.com/lhs
15     >
        <xsl:template match="paragraph">
            <xslcomponent:component
                name="summary1"
                class="com.sun.lhs.impl.samplecomponents.SummaryBuilder"
20             >
                <xslcomponent:args>
                    <xslcomponent:methodname="genSummary"/>
                    <xslcomponent:argname="bufferize" value="100"/>
                >
25             </xslcomponent:args>
            <xslcomponent:component>
                <xsl:apply-templates/>
            </xsl:template>
        </xsl:stylesheet>

```

30 The above example includes a Namespace declaration,
 xmlns:xslcomponent=http://www.javasoft.com/lhs,

invention. For example, the description refers to an external component instance, which will be recognized by those skilled in the art as an instantiation of an object in a system using object oriented language processing. However, other systems using other language processing mechanisms may be used. As yet another example, the described implementation includes software, but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.

calling the method associated with the external component defined in the style sheet; and

passing an argument contained in the style sheet to the external component for use by the method.

5 10. The method of Claim 9, wherein the step of loading the external component includes the step of locating an existing instance of the external component.

 11. The method of Claim 9 further comprising the steps of:
 naming a first and a second instance of the external component; and
10 processing the first and the second instance of the external component;

 12. The method of Claim 10 further comprising the step of processing different methods within the existing instance of the external component.

 13. The method of Claim 8 wherein the external component is defined to an XSLT processor and the style sheet using a Namespace.

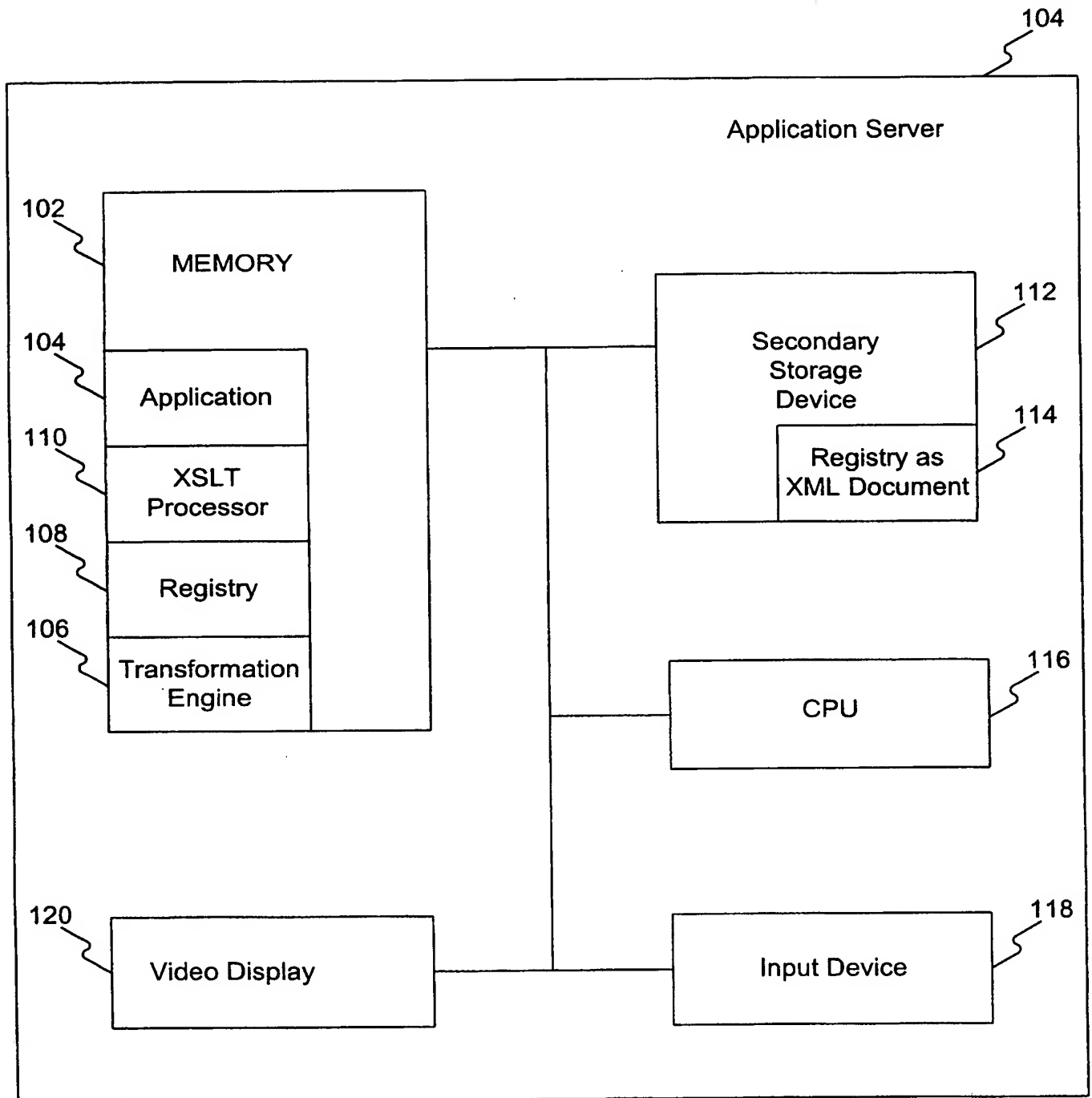
15 14. The method of Claim 8, wherein the step of processing the external component includes executing a method associated with the external component which generates a result and placing the result in a transform document.

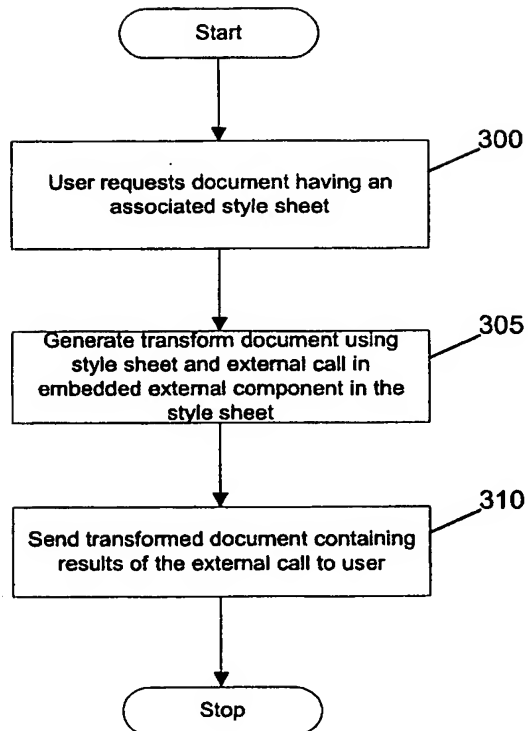
 15. The method of Claim 8 further comprising the step of providing a global context, a current input document fragment as processed by the style sheet and
20 a mechanism to generate a transformed document fragment to the external component.

 16. The method of Claim 8 further comprising the steps of
 retaining state information from the processing of the external component;
 reusing the state information during the processing of a second activation of
the external component, wherein a result generated by the processing of an external
25 component depends on the state information.

 17. A system for processing external components in a style sheet,
comprising:

 a style sheet containing a definition of an external component and a definition
of a method associated with the external component; and

**FIG. 1**

**FIG. 3**

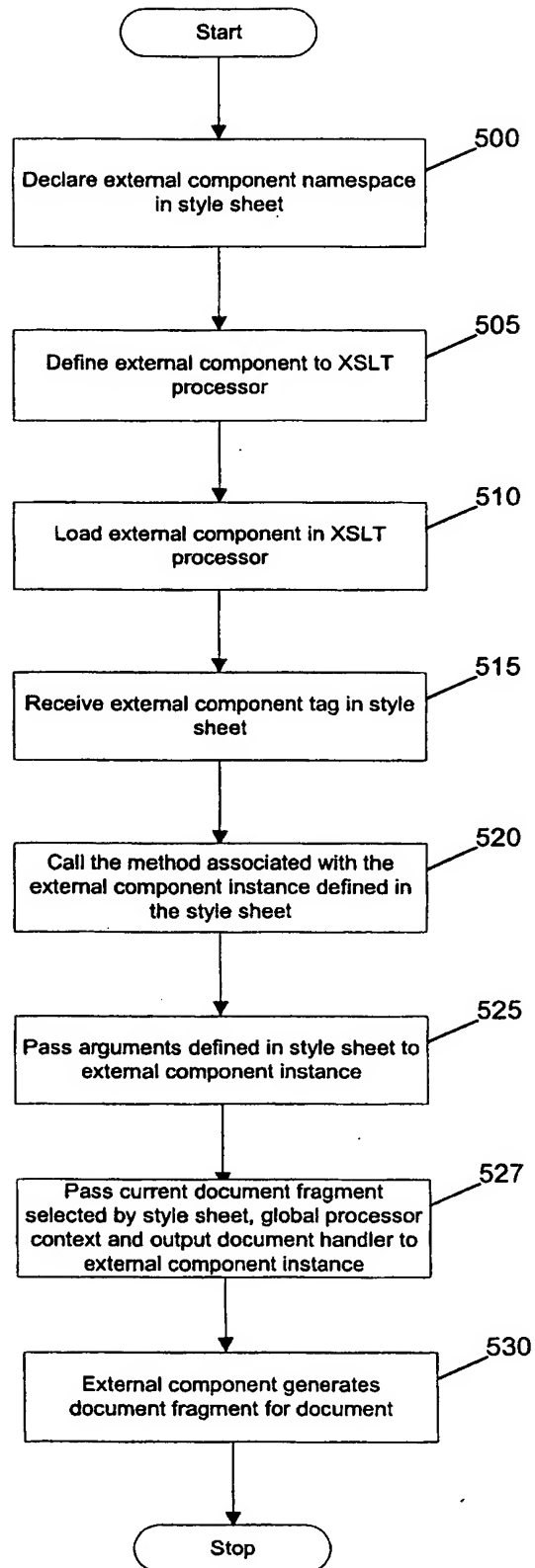


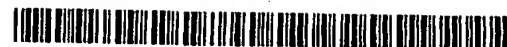
Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/06379

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ZEICHICK, A. XML and XSL, Network, November 1998, pp.23.	1-22
A	MCGRATH, S. Rendering XML documents using XSL, Dr. Dobb's Journal, v23, n7, pp..86(6)	1-22
A, P	US 6,031,989 A (CORDELL) 29 February 2000, ALL	1-22
A	US 5,860,073 A (FERREL et al) 12 January 1999, ALL	1-22
A, P	US 6,012,098 A (BAYEH et al) 04 January 2000, ALL	1-22
A, P	US 6,023,714 A (HILL et al) 08 February 2000, ALL	1-22
A	US 5,870,549 A (BOBO, II) 09 February 1999, ALL	1-22



(15) Information about Correction:

see PCT Gazette No. 25/2001 of 21 June 2001, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.